DEVELOPMENT OF AN URBAN SEARCH AND RESCUE ROBOT

by

Midshipman Bryan M. Hudock, Class of 2003
United States Naval Academy
Annapolis, Maryland

_____
(signature)

Certification of Advisers Approval

Associate Professor Bradley E. Bishop
Department of Weapons and Systems Engineering

_____
(signature)

_____
(date)

Assistant Professor Frederick L. Crabbe
Department of Computer Science

_____
(signature)

_____
(date)

Acceptance for the Trident Scholar Committee
Professor Joyce E. Shade
Deputy Director of Research & Scholarship

_____
(signature)

_____
(date)

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burder for this collection of information is estibated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burder to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (FROM - TO)
06-05-2003		xx-xx-2003 to xx-xx-2003

4. TITLE AND SUBTITLE
Development of an Urban Search and Rescue Robot
Unclassified

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME AND ADDRESS
US Naval Academy
Annapolis , MD21402

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME AND ADDRESS
,

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APUBLIC RELEASE
,

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The September 11th, 2001 bombing of the World Trade Centers in New York illustrated the many problems associated with rescuing the survivors of a collapsed building. The fact that those trapped beneath the rubble have low probability of survival after forty-eight hours, combined with the massive amount of debris that needs to be searched, make the urban search and-rescue mission extremely daunting. The solution to such a complicated problem lies in creating robots capable of quickly exploring a collapsed building and pinpointing the location of any survivors. The goal of this Trident Scholar project was to build an urban search-and-rescue robot. The primary goal was to develop a physical structure that would be unique and versatile enough to traverse a variety of terrain challenges. A design's effectiveness was judged on its ability to overcome the set of selected terrain types. The secondary goal was to produce a simulation that could be used to develop pre-programmed inputs: a 'fly-by-wire' style controller. This entailed creating an accurate model of the robot and its environment using a physics simulator. Using the simulation, various methods, including genetic algorithms, could be used to develop locomotive methods, or 'gaits,' for various types of motion (e.g., straight lines or parameterized curves). The movements could then be translated to the physical prototype and their effectiveness (the ability to move as predicted) analyzed. The end result of this project would be a robot with versatile mobility and a set of simulation tools. In the future, these simulation tools could be used to design and tune gaits for effective locomotion in a variety of terrain, with very little operator effort.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19. NAME OF RESPONSIBLE PERSON
			Same as Report (SAR)	47	Cornell, Elizabeth ecornell@dtic.mil
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			**19b. TELEPHONE NUMBER** International Area Code Area Code Telephone Number DSN

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39.18

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 6 May 2003	3. REPORT TYPE AND DATE COVERED

4. TITLE AND SUBTITLE
   Development of an urban search and rescue robot

5. FUNDING NUMBERS

6. AUTHOR(S)
   Hudock, Bryan M. (Bryan Michael), 1981-

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

US Naval Academy
Annapolis, MD 21402

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

Trident Scholar project report no. 311 (2003)

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
This document has been approved for public release; its distribution is UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT: The September 11th, 2001 bombing of the World Trade Centers in New York illustrated the many problems associated with rescuing the survivors of a collapsed building. The fact that those trapped beneath the rubble have low probability of survival after forty-eight hours, combined with the massive amount of debris that needs to be searched, make the urban search and-rescue mission extremely daunting. The solution to such a complicated problem lies in creating robots capable of quickly exploring a collapsed building and pinpointing the location of any survivors. The goal of this Trident Scholar project was to build an urban search-and-rescue robot. The primary goal was to develop a physical structure that would be unique and versatile enough to traverse a variety of terrain challenges. A design's effectiveness was judged on its ability to overcome the set of selected terrain types. The secondary goal was to produce a simulation that could be used to develop pre-programmed inputs: a 'fly-by-wire' style controller. This entailed creating an accurate model of the robot and its environment using a physics simulator. Using the simulation, various methods, including genetic algorithms, could be used to develop locomotive methods, or 'gaits,' for various types of motion (e.g., straight lines or parameterized curves). The movements could then be translated to the physical prototype and their effectiveness (the ability to move as predicted) analyzed. The end result of this project would be a robot with versatile mobility and a set of simulation tools. In the future, these simulation tools could be used to design and tune gaits for effective locomotion in a variety of terrain, with very little operator effort.

14. SUBJECT TERMS:
Urban Search and Rescue, robotics, evolutionary programming, genetic algorithms

15. NUMBER OF PAGES
45

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

ABSTRACT

The September 11th, 2001 bombing of the World Trade Centers in New York illustrated the many problems associated with rescuing the survivors of a collapsed building. The fact that those trapped beneath the rubble have low probability of survival after forty-eight hours, combined with the massive amount of debris that needs to be searched, make the urban search-and-rescue mission extremely daunting. The solution to such a complicated problem lies in creating robots capable of quickly exploring a collapsed building and pinpointing the location of any survivors.

The goal of this Trident Scholar project was to build an urban search-and-rescue robot. The primary goal was to develop a physical structure that would be unique and versatile enough to traverse a variety of terrain challenges. A design's effectiveness was judged on its ability to overcome the set of selected terrain types. The secondary goal was to produce a simulation that could be used to develop pre-programmed inputs: a 'fly-by-wire' style controller. This entailed creating an accurate model of the robot and its environment using a physics simulator. Using the simulation, various methods, including genetic algorithms, could be used to develop locomotive methods, or 'gaits,' for various types of motion (e.g., straight lines or parameterized curves). The movements could then be translated to the physical prototype and their effectiveness (the ability to move as predicted) analyzed. The end result of this project would be a robot with versatile mobility and a set of simulation tools. In the future, these simulation tools could be used to design and tune gaits for effective locomotion in a variety of terrain, with very little operator effort.

Keywords: Urban Search and Rescue, robotics, evolutionary programming, genetic algorithms

Acknowledgments

First and foremost, special thanks go to the project advisors, Associate Professor Brad Bishop and Assistant Professor Frederick Crabbe. Their constant willingness to commit time and effort in support of the project drove me to succeed, and I can't express my appreciation to them enough. Thanks next go to my parents, who have been a source of unwavering encouragement in all my endeavors. Thanks also to the employees at the United States Naval Academy machine shop: Tom Price, George Burton, Derrick Martin, David Guy, Dean Hockenberry, and others. Without their patience and their helpful suggestions, and their ability to produce the parts on the drawing board, this project would have never gotten off the ground. Thanks also to the Technical Support Division, specifically Norm Tyson and Joe Bradshaw, for their knack of finding solutions to the electrical challenges posed by this project. Thanks to my mother and father for their great patience and assistance. Thanks to my friends: Erik Phelps, Scott Sharrow, Mike Aurilio, Pete Glynn, Ethan Biter, Dave Crabbe, Dan Bailey, Bill Uffmann, Mike Mullen, Brian Abell, Colin Browning, Ken Byers, Max Desrochers, Jake Goldberg, Joe Hembree, Steve Kuehn, BJ Mahal, Nick Monti, Pauolo Singh, Patrick Hudock, Maria Guillily, Brooke Leach, and the many others who contributed by giving their time to help out or to simply listen to me complain. I can't fully express how lucky I am to enjoy life with such a great group of people. Last but not least, thanks goes to Professor Joyce Shade and the Trident Committee for their continued efforts in support of independent research efforts at the United States Naval Academy. To all, thanks again.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Recovering survivors from a collapsed building has proven to be one of the more daunting challenges that face rescue workers in today's world. Survivors trapped in a rubble pile generally have 48 hours before they will succumb to dehydration and the elements.[1] Unfortunately, the environment of urban search and rescue (USAR) does not lend itself to speedy reconnaissance or retrieval. The terrain is extremely unstable and the spaces for exploration are often irregular in nature and very confined. Though these challenges often make human rescue efforts deep within the rubble pile prohibitive, a robot designed for urban search and rescue would be very well suited to the problem.

Robots have already proven their worth in urban search and rescue, most notably in the aftermath of the September 11th, 2001 disaster (figure 1). The combined efforts of Professor Robin Murphy[2], a computer scientist at the University of South Florida, and Lt. Col (ret) John Blitch, culminated in the creation of the Center for Robot Assisted Search and Rescue (CRASAR), which coordinates and assists robotic search and rescue efforts.[3] Though these and other undertakings have shown the potential of a robot in the urban search and rescue environment, there is still room for significant improvement. Most notably, a unique structure and method of mobility could result in a vehicle much improved over existing urban search and rescue robots in terms of range and ability to overcome obstacles.

**Figure 1: Microtracs© Robot being used at World Trade Center Site
(http://www.csee.usf.edu/robotics/crasar)**

The primary goal of this project was to develop a robot that was highly mobile, lightweight, and easy to control. The robot was to be designed from the ground up with specific features to combat the irregular terrain found in an urban search and rescue environment. Specifically, the robot was designed to overcome three pre-selected obstacles that represented a sample of the types of impediments found in a collapsed building: irregular terrain, canyon, and ladder (figure 2). A chimney was initially selected as the third test obstacle. However, this was changed to a ladder obstacle when it was discovered that the ladder was used in standardized urban search and rescue test environments.
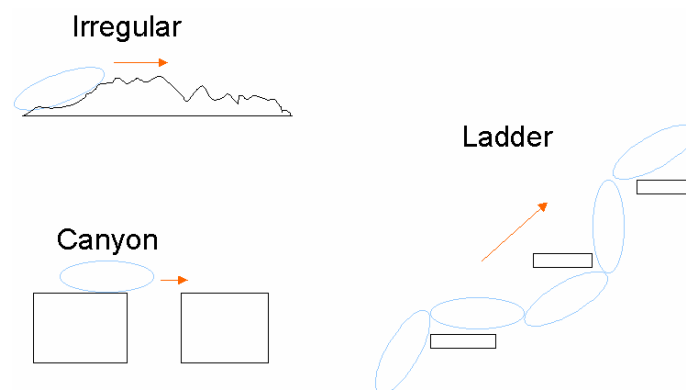
Figure 2: Selected obstacle types

The secondary goal of the project was to develop a simulation that modelled the physical prototype and its environment. This simulation could be used to produce pre-programmed controllers to simplify user input. This was achieved using a physics library to create a rigid body model of the robot and environment. Robots are distinguished by the level of tasks they can perform without human control. An autonomous robot is one that is capable of performing its mission without human assistance. At the other end of the spectrum is a robot that cannot function without an operator providing complete instructions. The pre-programmed inputs made the robot in this project semi-autonomous. The operator provided general instructions, such as 'move forward,' which the robot implemented by producing the specific motor commands. When fully combined, the efforts in physical structure and pre-programmed controllers yielded an easy to control robot, with unique structural features designed to enhance its mobility in the urban search and rescue environment.

2. PHYSICAL CONSTRUCTION - DESIGN

2.1. OVERALL MORPHOLOGY

The preliminary step of physical construction was to determine the robot's overall design. There were several factors that affected the overall structure. First, the robot needed to be able to move through the confined spaces often found in a collapsed building. Second, the robot needed to be very flexible in structure in order to easily maneuver over and around obstacles. Third, the robot chassis needed to be designed in such a way as to maximize strength while keeping overall weight as low as possible. Ease of transportation was an important factor for the robot, considering how the debris from a collapsed building could often make it difficult to access an area for search efforts. The final requirement was a versatile means of mobility. As the irregular

and unpredictable nature of the environment often called for creative ways of dealing with terrain challenges, a robot with a variety of methods of movement had a greater chance of succeeding in such a difficult setting.

The brief history of robotic involvement in urban search and rescue has seen designs that have almost exclusively relied on wheeled/treaded propulsion.[4] Though there have been alternative designs such as the snake effort by Choset (figure 3),[5] the versatility of a wheel in a variety of terrains combined with its simplicity and reliability have made it the common choice of mobility for search and rescue robots.[6] The addition of treads to a wheeled system has benefits and drawbacks. On the one hand the tread applies force all along its surface, in essence transforming the side of a robot into one large wheel. Though treads generally give robots an advantage in irregular terrain, the added complexity of a tread system and the need for specialized parts often make them prohibitive. In the case of an urban search and rescue robot, the benefits of a tread system in such an irregular environment outweigh the specialization and complexity problems that such a system adds.

Figure 3: Robot Design by Howard Choset (http://www.ri.cmu.edu/projects/project_407.html)

A snake-like robot is a second type of design that could be successful in an urban search and rescue environment. These robots consist of a series of linked cars, a design that has

significant structural flexibility. The concept of jointed cars makes snake-like robots highly

adapted to movement in confined spaces. Like an earthworm, a long and narrow robot can

traverse small openings and still retain a significant amount of mobility. There has been

significant past work with snake-like robots. In particular Saito, Fukaya, and Iwasaki, developed

the concept of the serpentoid curve, a model of motion of a snake-like robot using sine waves

that could be adjusted in amplitude, period, and phase.[7] This type of motion should prove to be

effective in certain types of terrain found in urban search and rescue.

Linked units have far more versatility and flexibility than a monolithic structure,

particularly when the links are powered. Powered links also provide the ability to perform shape

forming with the cars, another useful tool in achieving superior mobility. Unfortunately, when

the links of a snake robot are powered, the robot becomes inflexible and unable to adapt its

overall shape to the environment. A robot that conforms to the terrain is a far better design, as it

maximizes the amount of contact between the robot and the surface. In order to accomplish this,

the joints need to be compliant or able to maintain flexibility when powered. The concept of

compliance will be discussed in section 2.3.

A hybrid design that combines the versatility of the snake robot with traditional tracked

locomotion is a powerful design that should be quite capable in the irregular environment of a

collapsed building. Keeping the cars identical in nature simplifies the design and construction,

and also keeps the overall design physically flexible. Mounting treads on both the top and

bottom of each car in the vehicle enhances mobility in confined areas by allowing the robot to

use the roof of the environment along with the floor in order to propel itself forward. Each car

should be connected by a link that is powered and has two degrees of freedom. The links should

have compliance, the ability of two joint connected bodies to flex and move while still being

attached to the link. This maximizes the robot's ability to conform to the ground. The resulting design should have the physical ability to overcome the three types of test obstacles.

This design should be very effective against irregular terrain (figure 4). Treads offer a large surface area that can be used to contact the floor and generate normal force for movement. This is advantageous in irregular environments, as treads continue to generate force even when the terrain becomes uneven and the points of contact vary in terms of relative location to each car. Additionally, the selective compliance of the joints keeps the robot conformed to the ground and maximizes contact between each car and the ground. This configuration optimizes movement of the robot in the desired direction of locomotion. In case the treads become ineffective, the robot should have the ability to inch itself along using the motors in the car links to generate a caterpillar motion. Finally, the robot should be capable of serpentoid motion, which might prove successful in certain types of irregular environments. The tread design will be discussed in greater detail in section 2.2.
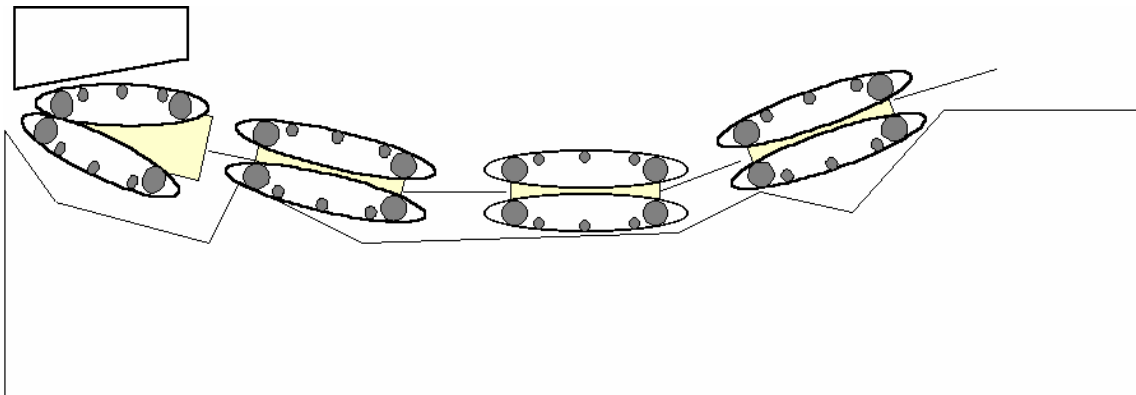
Figure 4: Jointed robot conforms to irregular terrain, maximizing contact

The robot should also be effective against a canyon. The use of a powered, two degrees of freedom joint to link each car gives the robot a vertical mobility that should be useful in overcoming horizontal gaps. The link design will be discussed in much greater detail in section

2.3. Using algorithms such as the one developed by Nilsson[8], the robot could be manipulated to achieve an upright position. From this position it could then lunge across the canyon and use the terrain on the other side to pull itself over the gap.

A robot of this nature should be well suited to surmounting a ladder obstacle, in which the robot's only method of traversing would be to wind its way through the rungs. The robot's ability to use treads on both its top and bottom should allow it to maintain forward motion by using the rungs above and below it for contact. Pitching the car upwards with the powered links should provide the force necessary to maintain contact with a rung located above a car. The robot's unique structural features should give it a large advantage when trying to defeat this difficult obstacle.

2.2. CAR DESIGN

Having chosen the overall morphology, the next design decision is to determine how many faces of each car should mount treads. Assuming each car is a rectangular prism with the (+x) axis in the direction of motion (figure 5), the first set of treads should contact the floor, along what would be considered the (-z) face. This arrangement is the structure that almost all existing treaded robots use. However, treads on the (+z) face could use contact with the roof of the environment to propel the robot forward, significantly improving mobility in confined spaces (figures 6,7). Treads on the top and bottom of the car utilize a design similar to the pyramid-exploring robot built by the iRobot company.[9] The success of the pyramid-exploring robot in traversing through a long, narrow tunnel validates the tread placement design concept.

Treads on the sides (+,- y) would further enhance mobility by using side contacts for forward motion. Side treads would make the orientation of a car almost irrelevant, because it

would not matter which face of the car was facing down in order to use the floor for movement. Adding treads on more than two faces would require a support structure that is extremely complicated. These added structural and design complexities would outweigh the benefits of having treads along the sides of the vehicle. Consequently, the most realistic car design would call for treads on only the (+z and –z) faces.
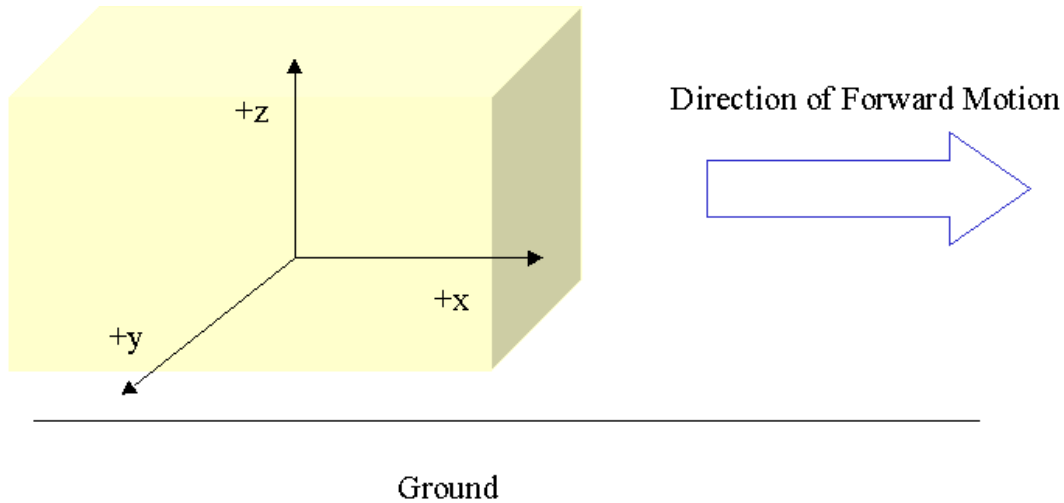
Figure 5: Relative Car Axis
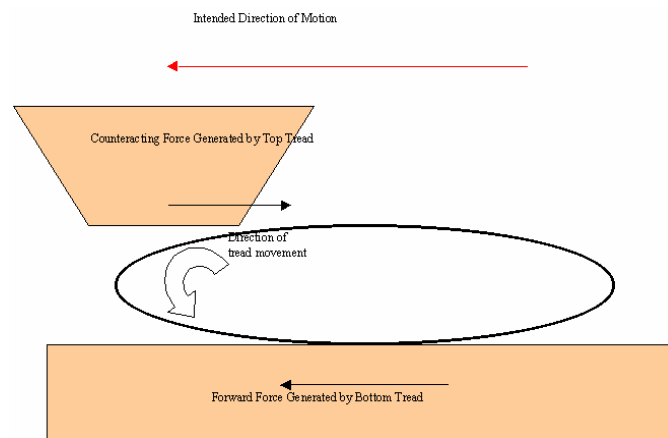
Figure 6: One tread system works against itself in confined spaces
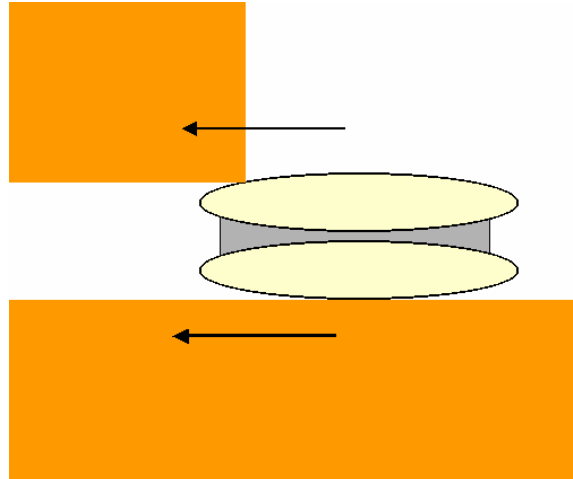
Figure 7: Two tread sets enable robot to use floor and ceiling for foward movement

Another consideration for car design is whether the faces with treads should have one large tread or two smaller treads. A one-tread design should be much simpler and more economical in regard to cost, weight, and electronic complexity. However, a face with one tread can only generate forward or backwards movement. Though two treads are more complicated, it gives each face differential drive. Differential drive is the ability to power the wheels on the left and right sides of the car at different rates. With differential drive, each car should have the capability to change its angular orientation along with its position simply by adjusting the speeds of the two wheels.

Controlling a segmented robot with differentially driven cars is a challenging task. Curve fitting is a process that uses controlled differential drive to move the vehicle in arcs described by a derived radius of curvature.[10] In the case of a two-wheeled (or treaded) robot with given angular velocities for the left and right wheel (ω_R, ω_L), and a wheel radius of (r), the overall speed of the robot is governed by the following equation:

$$\| V \| = (\omega_R + \omega_L) * r/2$$

Additionally, the radius of curvature (R) is determined by the following equation, where (d) is the axle distance between the right and left wheel. :

$$R = [2*(\omega_L + \omega_R)] / [d*(\omega_L - \omega_R)]$$

The benefit from this type of motion control applied to each car in the robot outweighs the complexity and cost that two treads would add. Consequently, each car has a set of two treads on the top and bottom face of the vehicle.

2.3. LINK DESIGN

One of the most important considerations in the physical design is the configuration of the links between cars. The number of degrees of freedom (DOF) that each link enjoys is one of the critical characteristics in the link design (figure 8). In order to perform simple turns about the z-axis (yaw), the links on the robot need to employ one degree of freedom. Though this would fulfill minimum functionality, one degree of freedom makes the robot essentially planar. An urban environment will call for significant movement in all three dimensions. The ability to rotate about the y-axis (pitch) is essential for optimizing mobility. In particular, the ability for each car to pitch up and down is necessary for surmounting a ladder obstacle. Additionally, a link that is flexible on two axes of motion improves the robot's mobility by allowing it to move in a variety of ways, simply by employing the link motors alone. It is clear that two degrees of freedom is at the point of diminishing returns, in terms of weight and cost versus enhanced

mobility. Though the advantage of a two DOF link is patently obvious when compared to a one

DOF design, adding a third degree of freedom about the x-axis (roll) does little to benefit the

overall mobility of the robot. The addition of a third motor to the link would significantly

increase weight, cost, and structural complexity. Consequently, the link between cars should be

powered in two degrees of freedom (yaw and pitch).
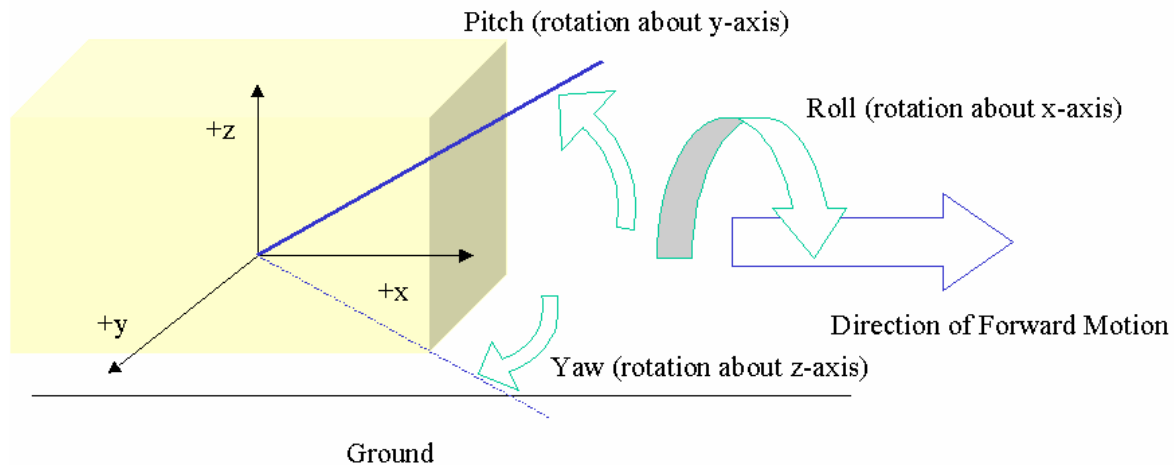
Figure 8: Angular orientation of car

Each car link needs motors that can accurately determine their own shaft position. This

gives each motor in the link between cars the ability to determine the current relative angles

between cars. In addition, it allows the user to specify new pitch and heading angles to which

the motors move. The most efficient means of moving the robot forward is to have each car

aligned in a direction tangent to the path of motion. Any variance in the relative angle between

cars will cause them to work against each other, preventing effective overall movement. With a

passive link, the slightest disturbance causes the cars to go out of alignment. Without the ability

to correct the error, the problem will only increase with time until the robot is unable to move,

similar to a jack-knifed big rig.

Not only does the ability to control relative angular position make simple movements more efficient, it realizes the potential for shape forming, in which the cars are aligned into specified curves and lines. For example, arranging the cars in a straight line will produce overall movement in the direction of the line. Arranging the cars in an arc will cause the robot to turn about a point of rotation that is determined by the angle between the cars. Beyond simple geometric shapes, shape forming can be used to implement the serpentoid curve and other more complicated motion patterns. As such, the link between cars has to be powered and not passive.

The ability to control pitch gives a segmented robot the capability to perform relative shape forming in the vertical plane as well. Nilsson established an algorithm to maximize the length of a linked robot that can be moved upright from a horizontal position.[11] This algorithm was based on an ability to pitch the links up to 90 degrees in relation to the proceeding car. Maximum pitch angle in each link between cars is the biggest factor in determining the torque required to lift multiple cars (figure 9). As can be seen from the figure, cars work with or against the lifting torque produced by a joint depending on which side of the lifting action they are on. Those cars on the lifting side of the line work against the torque the motor is producing. Those cars that are moved across the line aid the motor in lifting the cars. Therefore, pitch range of motion becomes the critical factor in lifting. As the angle that each car can be elevated increases, the corresponding torque required to lift multiple cars decreases.

Maximum Pitch Angle (degrees)	Number of cars Working against lifting	
45	5	
60	4	
90	3	
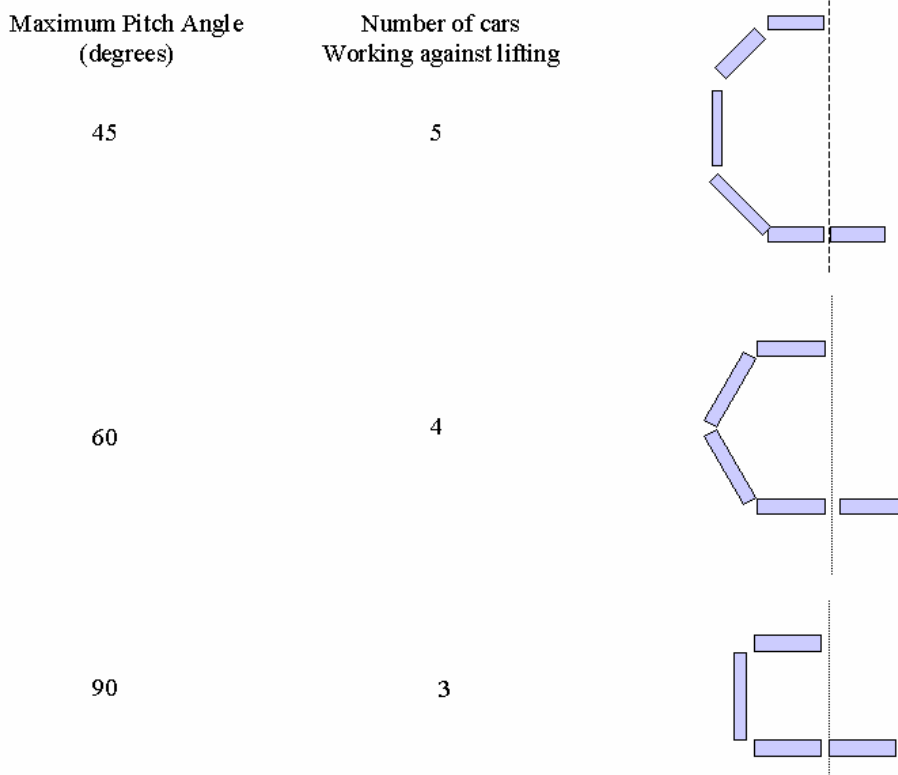
Figure 9: Torque requirements depedning on maximum pitch angle

Links connecting directly to a motor shaft are rigid in nature, because the cars are locked into the angular positions specified by the operator. The idea of compliance, or "give" in a link, becomes an important consideration for improved mobility. This factor also determines how a vehicle responds to irregularities in its surface contact. For example, a car whose wheels are rigidly attached to the body has no compliance. When the front right tire of the car contacts an obstacle, the entire right side of the vehicle lifts up as the wheel climbs over the obstruction. If a car has four-wheel drive, it loses the power provided by the right rear tire. Adding passive compliance to the vehicle structure is a potential solution to this problem. Passive compliance means each wheel is attached to an independent axle that can translate in the upward (or downward) direction. When the front right wheel contacts an obstruction, it translates upward,

independent of the body as it climbs over. The other three wheels remain in contact with the ground.

Though passive compliance is more effective in dealing with irregular terrain, it does not allow for pitch control in a jointed robot. Selective compliance allows the user to adjust the level of compliance based on the expected situation and environment. A stick and a chain illustrate both aspects of the compliance concept. A stick is an object with no compliance. The relative position of every section of the stick is always constant, but it has no ability to conform to the environment. A chain conforms to an irregular environment, but it is impossible to control the position of each link in relation to the others. In this way, selective compliance seeks to combine the abilities of a stick and a chain. A known example of selective compliance is a magnetorheological (MR) fluids shock absorber. The nature of this fluid is such that its compliance can be electronically adjusted thousands of times per second, thus constantly adapting to a changing environment to provide optimum performance.[12] Obviously, an MR fluid shock absorber is something much too complex for a lightweight robot. However, the concept of selective compliance is critical for creating a car link that can control pitch while maintaining the mobility of the robot.

Selective compliance can be achieved in a robot without a complicated physical system like an MR shock absorber. The goal of selective compliance is to keep as many cars in contact with the floor surface as possible, especially over irregular terrain. Achieving this goal involves modifying the portion of the link that controls pitch. Instead of a rigid connection from the motor controlling pitch to the next car, the two entities are connected by a rod, which has a rigid connection on one end and a pin joint on the other. Springs are connected on the top and bottom

of the motor shaft, with the other ends of each spring connected to the top and bottom of the back

of the car (see figure 10).
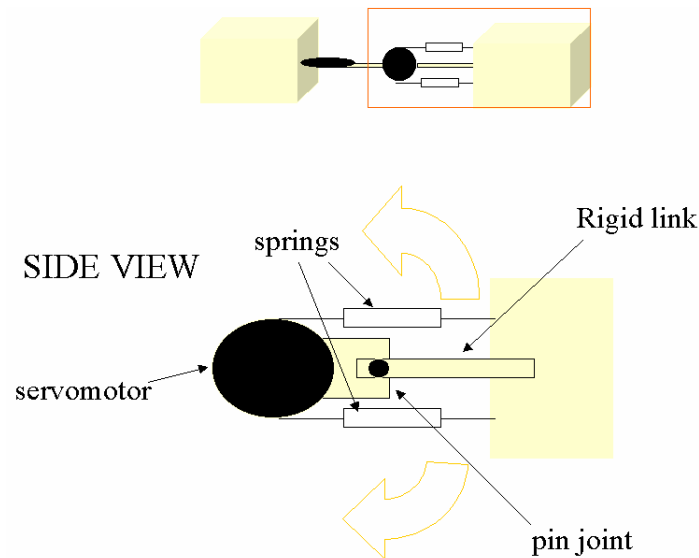
Figure 10: Diagram of servomotor controlled pitch link

Rotating the motor shaft changes the equilibrium position of the car. If the motor shaft rotates

counter-clockwise, the equilibrium position of the car rises, and the car lifts. If the motor shaft

rotates clockwise, the equilibrium position of the car lowers, creating the opposite effect. The

spring-connected joint should also have a large degree of flexibility, especially when compared

to a rigid link. When enough force is applied to the body of the car, it should stretch the springs

in the link and shift its position. When this force is removed, the springs return to their

equilibrium and the car moves back to its original position. Lowering a car's equilibrium point

past the floor creates a normal force on the ground greater than the weight of the car. Assuming

the car is on a flat surface, the addition by the link to the normal force experienced by the car is

described by the following equations. As described in Hooke's law, the force exerted by an

extension spring is a function of its spring constant (k) and the distance it is stretched (x):

$$F = k * x$$

Specifically, the force exerted by an extension spring stretched around a motor shaft is described by the following equation, where (L) represents the horizontal distance from the pivot point to where the spring is attached on the front car, (r) represents the radius of the motor shaft wheel around which the spring is wrapped, and (θ) represents the angular position of the motor shaft:

$$F = (L + \theta * r) * k$$

In order to keep a car from falling, a spring has to provide a moment around the pivot point equal to the moment produced by a car mass (M) located a horizontal distance (d) from the pivot point:

$$(L + \theta * r)*k*r = M*g*d$$

If the parameters of the spring produce a moment greater than $(M*g*d)$, then the spring is capable of elevating a car. Conversely, rotating the servomotor the opposite direction added the spring's force to the weight of the car. This increases the normal force (N_f) applied by the floor to a tread:

$$N_f = (L + \theta * r)*k + M*g*d$$

The ability of a tread to produce forward motion is governed by the frictional force it can produce by contacting the ground. With the addition of the spring force, where (μ) represents

the coefficient of dynamic friction, the maximum friction force between the tread and the ground

becomes:

$$F_f = ((L + \theta * r)*k + M*g*d) * \mu$$

This increased normal force means a greater frictional force produced by the tread and thus

greater traction. The normal force added by the link is a function of the equilibrium point of the

car in relation to the point of contact with the tread. When a car encounters a body that forces it

upward, the equilibrium point of the car shifts farther down relative to the ground. This causes

the spring link to stretch, resulting in an increased normal force on the tread. At the same time,

the springs have a built-in compliance, which helps keep as many cars as possible contacting the

floor surface when surmounting obstacles. The tensioning of the spring link needs to be

carefully controlled. As the link applies force on both cars to which it is attached, too much

torque could cause the cars to lift and coil up, defeating forward motion by the robot.

3. PHYSICAL CONSTRUCTIONS – RESULTS

3.1. OVERVIEW

The final robot consisted of three cars, which was the minimum number needed to

accomplish basic link actions such as pitching a single car up or down.   The cars were linked to

each other by two-dimensional links (see figure 11). Each car was constructed out of ¼" thick

plywood and was reinforced by ¼" steel threaded rod. The total weight of the robot was

approximately thirteen pounds. Each car had four treads that could each be independently

powered. Both cars were attached with the spring based two-dimensional link. The car links had

a maximum pitch angle of approximately forty-five degrees, and a heading range of motion of fifteen degrees in either direction.

**Figure 11: Final robot prototype**

3.2. TREADS

The treads would never have left the design table if parts could not have been found that met physical requirements and were within a respectable price range. The treads needed to be strong, durable, and conducive to modification according to the dimensional requirements of the robot. Finding treads that met these requirements proved a much greater challenge than expected. The initial idea was to seek out molded rubber treads of the type found on many R/C model vehicles. Unfortunately, manufacturers could not be found that would custom make rubber treads at a low cost, so an alternative needed to be found. Timing belts manufactured by W.M. Berg proved to be an acceptable solution. The belts were designed to be "no walk," or grooved in more than one direction, such that they would not slip off the side of the pulley. Unlike most timing belt pulleys, the W.M. Berg pulleys had no flanges that would interfere with

the contact between ground and tread. The belts consisted of steel cords surrounded by

polyurethane, making them very strong and durable. Since the belts come in bulk length, the

necessary size for each car's tread were cut and spliced from the ordered length according to

design specifications. The price for the belt and pulleys was not prohibitive. Consequently, the

W.M. Berg parts were selected to serve as the wheel/tread system for the robot.

The W.M. Berg timing belt used for the treads displayed some benefits and some

drawbacks (figure 12). The "no-walk" system proved to be as good as advertised, as the treads

would stay aligned to the timing belt pulley without the aid of flanges that would interfere with

ground contact. The first downside to the timing belt emerged when it came time to splice pieces

together into loops to be used as treads. W.M. Berg offered a splicing kit for such purposes.

Unfortunately, the manufacturer's splicing system left a large gap in the tread that would have

caused unacceptable snags on the contact surface. The splicing kit itself was also very

expensive. Consequently, the decision was made to explore other splicing options. Boring small

holes at each end of a piece of timing belt and using steel thread to sew the two ends together

proved to be a solution. This method proved usable, but it was not optimal. This was mainly

due to the thread interfering with the grooves that held the timing belt to the timing belt pulley.

By carefully selecting the points at which the splicing occurred, the impact of the steel threads

could be minimized but never eliminated completely.

Surprisingly, the strength of the timing belt proved to be another drawback when it was

used as tread. When looped around small objects such as the wheels of the robot, the timing

belt's steel cord center acted as a spring that gave the belt a natural tendency to straighten out.

This tendency of the timing belt to push outwards further enhanced the normal force contact with

an object. However, this characteristic also lessened the contact area between pulley and belt,

making it more likely a tread would be thrown, rendering one side of the car useless. A less rigid

material under tension would sag and ensure greater consistency of motion.
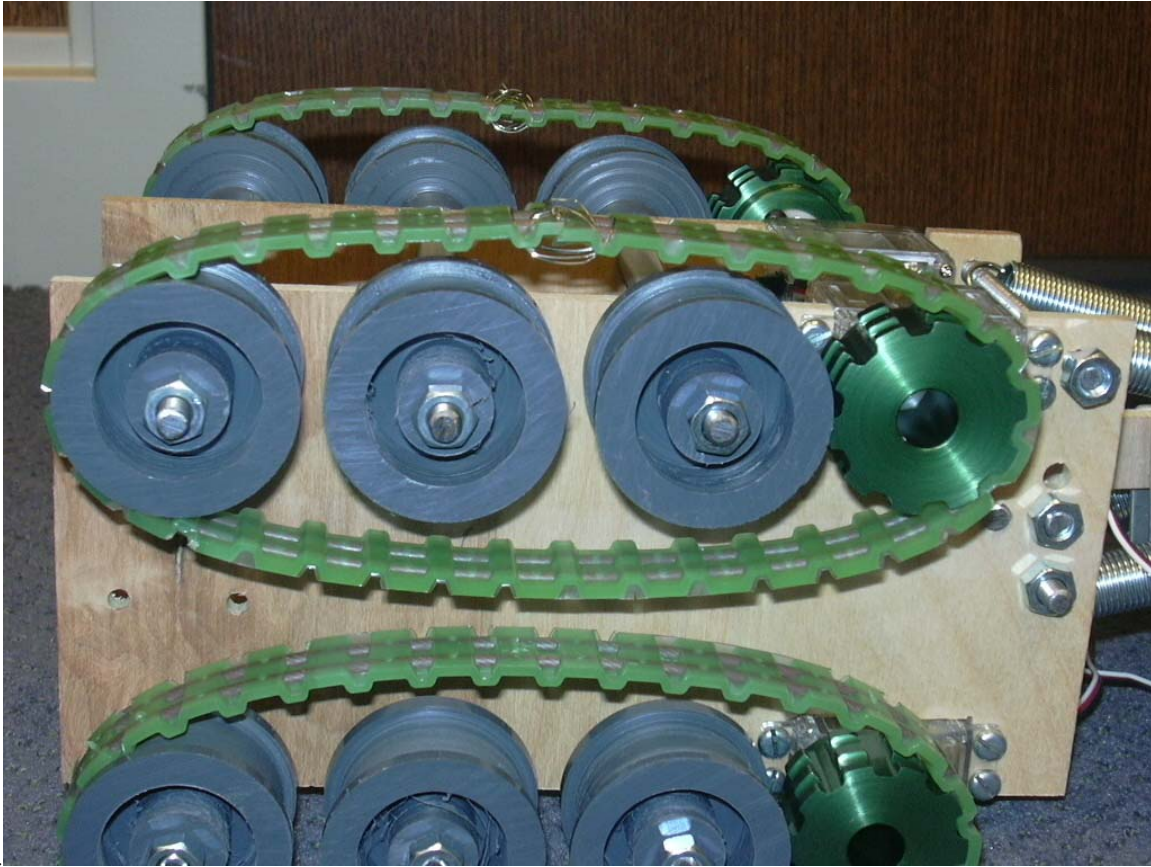
Figure 12: Tread System using W.M.Berg "no-walk" timing belts

The tread system included passive idler wheels that were mounted alongside the wheels

actually driving the treads. By maintaining contact points and normal forces along the entire

length of the bottom of each tread, the idler wheels were the parts that gave the tread its high

capability in terms of mobility. Thus the idler wheels needed to be lightweight, but also of a

material that could be grooved to provide a continuous path for the tread to follow. PVC plastic

was selected for its lightweight yet durable nature, as well as the ease with which it could be

lathed and shaped. The idler wheels' unique shape minimized weight while still allowing them to fulfill their roles of guiding the tread and providing contact points (figure 13).

Figure 13: Custom built idler wheel

3.3. MOTORS

The robot design required two different types of motors; one type of motor to power the treads and one type to power the links between cars. Since the power source of the robot was DC current, both motors needed to be of the DC variety. Both types also needed to be lightweight motors capable of producing high torque. The one significant difference between the two motor types was that the link motors required position control, while the tread motors needed only speed control. A heavy-duty R/C servomotor was able to accomplish both tasks. Servomotors were perfectly suited to powering the links because the built-in shaft encoders provided accurate positional control. The one drawback to using a servomotor to power the treads was that servomotors have mechanical stops that only allow a range of motion of approximately 180°. Additionally, the feedback loop of the servomotors is controlled by a potentiometer connected to the output shaft. This potentiometer had to be removed or isolated

from the circuit, and a resistor bridge put in place to allow for continuous running. With these modifications the servomotor could be used to run continuously.[13] The motor selected was a CS-80MG Pro, made by Cirrus. Weighing only 2 ounces, it is able to deliver 129.8 oz-in of torque at 6 volts. Constructed with metal gears and ball bearings, the servomotor is much more durable than typical servomotors that use plastic gearing.

Overall, the servomotors performed exactly as expected. They were easy to control, were easy to mount to the chassis, and despite their small size, fulfilled all the torque demands of the tread and link systems. The process of modifying the servomotors for continuous running proved to be the biggest drawback. The procedure involved modifications to the physical structure and the control board of the servomotor. Though these changes were very easy to understand, modifying each servomotor involved prolonged amounts of tedious work and consumed significant amounts of time.

3.4. CHASSIS

In order to optimize the robot's effectiveness, the chassis of each car had to fulfill a few important requirements. First, the chassis needed to be lightweight. This was important not only to increase the ease of transportation, but also to assist in the implementation of a two-degree of freedom link. Assuming a distance of about four inches from servo shaft to vehicle center of mass, a single servomotor was capable of lifting a car that weighed approximately 30 ounces. Making the chassis as light as possible ensured that the link would be successful in being able to elevate a car. The chassis had to support three items: the servomotors driving the treads, the idler wheels, and the link structure. One of the benefits of using servomotors to drive the wheels

of the vehicle was that the servos could be mounted on a thin piece of material, since the servomotor casing had fabricated holes that allowed it to be bolted in place.

The chassis design eventually selected consisted of two flat sidepieces identical in appearance. Threaded steel rod was run through holes in the two sides. The rods supported the idler wheels while allowing them to freely rotate. Aluminum spacers were placed on the rods to reinforce the sidepieces while maintaining the car's exact dimensions. Aluminum and wood were used to construct the various car prototypes, as they were both adequately strong while costing little in terms of weight. The ability of the aluminum spacers to strengthen the sides made the two materials relatively interchangeable, with wood being used to construct the sidepieces for the chassis of the final robot.
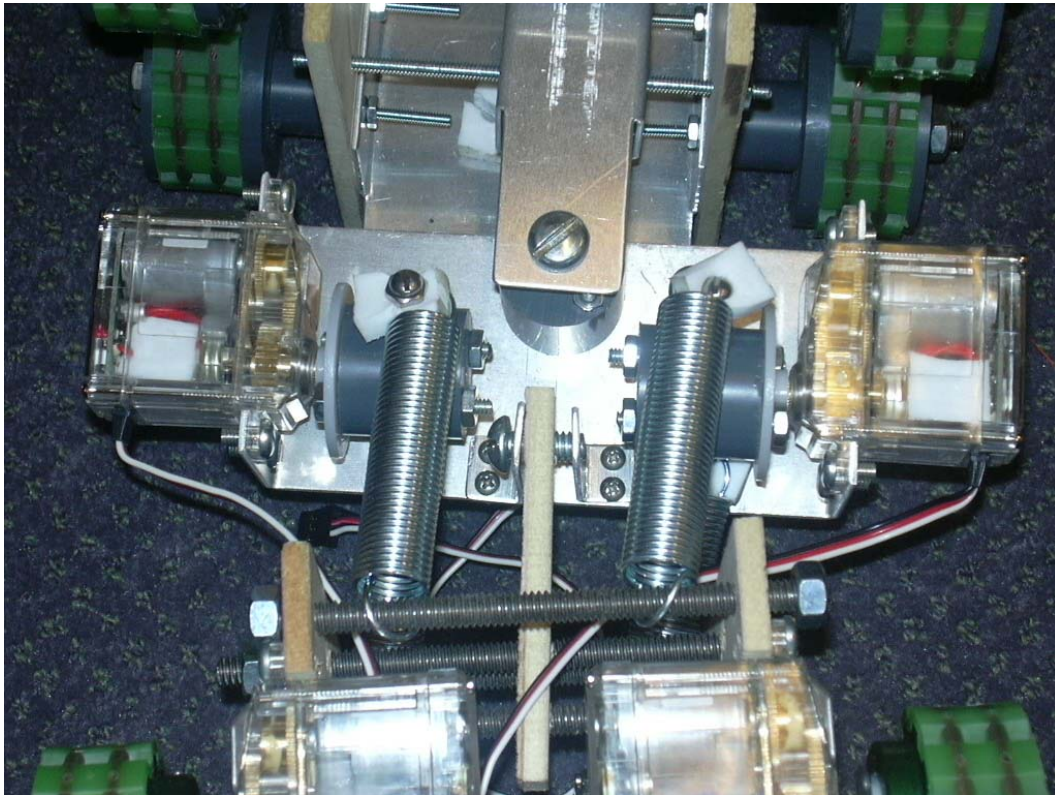
3.5. CAR LINK

A mock-up was built to achieve proof of concept that the spring-link system worked as designed. It consisted of a single servomotor connected to a lighter chassis. The link on the mock-up successfully raised the car. By rotating the pitch servomotors in one direction, each spring provided a combined force that elevated the front car. Additionally, the springs were compliant enough to accommodate a significant deviation from the equilibrium position by the car in the up or down direction. This demonstrated the ability of the car link design to control pitch, while at the same time allowing the cars to conform to outside forces.

The two-degree of freedom links used to connect the cars expanded on the design concepts established in the mock-up (figure 14). The link initially consisted of four pieces. The first was a bracket that attached to the servomotor that controlled the car's heading to the chassis of the rear car. Mounted on the wheel of the heading servomotor was a second bracket that held

the two servomotors controlling pitch. The final two brackets were used to connect a passive wooden pivot to the rear of the front car. The problem with this initial design was that elevating the front car placed enough torque on the heading servomotor shaft to bend it significantly. The bending of the shaft caused the link to tilt forward and misalign the pitch pieces and also posed the risk of breaking the shaft completely. Consequently, two additional pieces were added to transfer the torque from the servomotor shaft to the chassis of the rear vehicle. A plastic cylinder with a hole in its center was bolted onto the heading servomotor wheel. A bolt was inserted into this hole and held in place by a bracket positioned above the cylinder. The bracket holding the bolt was attached to the rear car chassis. As a result, any impulse of the link to bend forward would be checked by the bolt-cylinder arrangement. Instead of the servomotor shaft bearing the torque, it was transferred to the chassis of the rear car.

Figure 14: Close-up of two DOF Car link

3.6. CONTROL MODULE

The goal of the control module was to create a semi-autonomous system. This meant that the user gave simple commands such as "forward" or "pitch up," which triggered the microprocessor to issue pre-programmed motor commands that accomplished the action. A prototype control module was constructed to enable simplified user input and also eliminate the need for a desktop computer to interface between robot and operator. A Rabbit microprocessor was used for the highest level of control after the user. Each car had a Pontech SV-203 servomotor controller board that directly controlled the servomotors on the car. The Rabbit in a master-slave configuration controlled the three Pontech boards. The operator interface consisted of a series of switches. Flipping a switch sent an input signal to the Rabbit, which then selected the appropriate sequence of commands to pass to the Pontech boards via serial connection. The Pontech boards then provided the direct servomotor control to accomplish the motion. Forward motion, reverse motion, and elevation of the front car were all programmed into the Rabbit as pre-programmed motor sequences that could be selected by the operator.

4. FUTURE WORK – PHYSICAL DEVELOPMENT

The initial effort in future work on the physical development of the robot would involve thorough testing of the prototype's capabilities. In particular, analysis of the robot's ability to overcome the pre-selected obstacle types would be critical in determining the overall effectiveness of the design concept in the urban search and rescue environment. Areas of the current physical design could be improved, particularly the treads and their drive motors. Expanding the robot to six cars would give it an even greater capacity for improved mobility. A six-car robot would have the ability to implement Nilsson's vertical shape forming algorithms as

well as serpentoid motion. The flexibility of the robot could be further improved by mounting

treads on the sides as well as the top and bottom. Such a structure would make the roll

orientation of the robot unnecessary for controlled movement. Specializing the front car, by

having it taper to a point could also improve the mobility of the robot. The treads would also

taper, starting far apart and ending at the front of the car close together. This type of car would

be very well adapted to burrowing through loose rubble.

As designed, the robot needs a tether to connect it to its operators, both for power and the

two-way communication required of a semi-autonomous vehicle. A structure mounted to one of

the cars and capable of holding a significant amount of tether cable would have to be designed to

allow the robot to operate for a significant distance from the operator. Finally, the robot's

capabilities could be fully tested in an obstacle course containing a variety of obstacles including

a canyon and stairs. These efforts would culminate in a robot optimized in terms of structure and

mobility for urban search and rescue.

Beyond mobility and structure there are additional areas of urban search and rescue

robotics that can be explored. Power sources can be examined in an effort to find one that would

allow the robot to retain its lightweight nature while also giving it the capability to operate

without a tether. Sensors such as Global Positioning System (GPS) can also be examined for

enhanced effectiveness in the urban search and rescue environment. In all of these cases, the

current robot prototype could serve as a test bed for these efforts.

5. SIMULATION DEVELOPMENT

5.1. INTRODUCTION

For a simple remote-controlled car, an operator has little problem directly controlling the few motors that govern speed and steering. In the case of a snake robot, where each car is equipped with several independent motors, control complexity for the operator becomes an issue. For a three-car robot in which each car is connected by a three-servomotor link, an operator has to operate eighteen motor inputs more or less simultaneously in order to achieve effective real time control. Even simple overall movements such as moving forwards or backwards or turning the vehicle require a great deal of operator involvement. Actions such as relative shape forming are too complex and time consuming for a manual operator. The operator attention required to directly control the motors distracts the operator from the main purpose of the robot, which is examining and exploring the environment of a collapsed building. Additionally, effective control of the motors via direct manipulation requires a skilled operator who is knowledgeable of the robot and the working commands. Pre-programmed motor inputs decrease the complexity of control to the point where specialized personnel are not required to use the robot. Instead of having to constantly input 18 different values, the operator can command 'turn left' and the pre-programmed inputs coordinate the various motors so that vehicle turns left. As such, simplifying operator input to the robot becomes a necessity for effective operation.

There are two primary methods for developing pre-programmed motions. The simplest method is an intuitive approach, in which the operator experiments with direct control of the robot until a desired behavior is achieved. The sequence of operator activated motor inputs is then saved for use as a pre-programmed response for that particular behavior. The intuitive

method can be good for simple kinematics problems such as forward movement of the robot. However, the intuitive approach for more complicated motion problems has several disadvantages. First, a fully functional robot is required before testing can begin, and any breakdowns by the robot would disrupt the developmental process. Second, each individual test requires a significant amount of preparation and set up time. Though it is possible to develop pre-programmed inputs for a complicated movement using only trial and error, there is no way to ensure that the solution is optimal for other considerations such as power consumption or time.

Inverse kinematics is the other mainstream method of motion determination. This process uses equations to determine the joint angles needed to achieve a given end point. However, the inverse kinematics of a problem becomes increasingly complex with additional joints. A multi-jointed robot like the one built in this project has an infinite number of inverse kinematics solutions, and does not lend itself to easy solutions using this method.[14] In order to produce successful sequences of motor commands to implement complicated movements, alternatives to the intuitive and inverse kinematics approaches need to be explored.

Genetic algorithms are a method well suited to the problem of developing pre-programmed inputs for complicated motions. Genetic algorithms involve breaking down a computer program into a sequence of numbers referred to as a chromosome, where each component in the chromosome is known as a gene. A group of individuals with these chromosomes are randomly generated and tested using a simulation. Upon completing a run in the simulation a chromosome is assigned a fitness, which is a grade of the performance based on user specified guidelines. Genetic algorithms closely mirror the biological process of evolution. The individuals with the best chromosomes are used to produce a new generation of individuals in a form of sexual reproduction. In what is known as a "one point crossover," [15] two parents

are selected to create a new individual. Up to a randomly selected point, the new chromosome is made using the first parent's genes. After the point, the other parent's genes are used to complete the chromosome. Mutation is used to randomly modify genes. What makes the process successful is the computer's ability to create and test extremely quickly. Hundreds or even thousands of cycles can be run until acceptable results are achieved (figure 15). In this case, the chromosome is a sequence of motor commands for a set period of time. The simulation is a program that can accurately model the robot and its surroundings. If trying to optimize forward movement, the fitness of a chromosome can be the distance from the starting position the robot has moved after receiving the set of motor inputs.
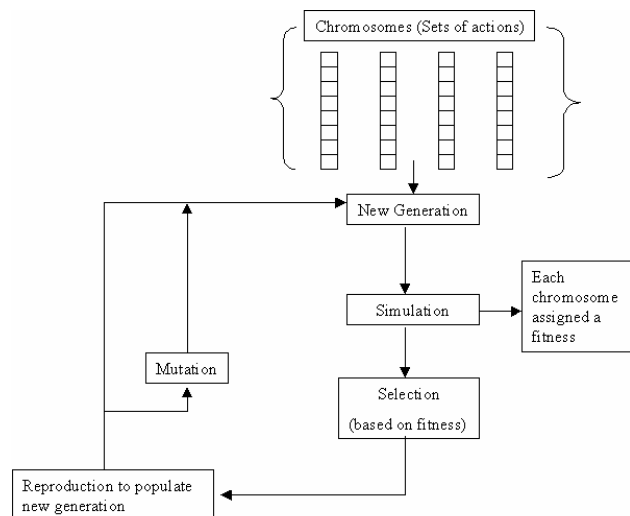
Figure 15: Breakdown of genetic algorithm

In terms of developing successful pre-programmed motions, genetic algorithms have several advantages over the intuitive approach. The fact that the entire process can take place in simulation means that a fully operational robot is unnecessary in the efforts to develop effective pre-programmed controllers. Second, simulations are easily adaptable to a change in the environment or the robot. A change in the robot's dimensions or the height of an obstacle might take hours to implement in the physical environment, while modifications to the simulation are

instantaneous. The ability of genetic algorithms to develop usable pre-programmed motions from randomly generated sets gives the process the potential to solve kinematics problems that cannot be described through longhand kinematics equations. A final advantage lies in the ability of the user to define the fitness parameter used to compare chromosomes.   Pre-programmed inputs can be optimized for time, power consumption, relative car position, or even a combination of several factors. All these advantages make genetic algorithms a powerful tool in the development of pre-programmed inputs.

Several steps had to be accomplished before genetic algorithms could produce optimal pre-programmed inputs. The first step was to develop a simulation that would be capable of supporting the rigid body dynamics of the robot and its interaction with the environment. Next, the motor commands needed to be formatted into a chromosome that could be evolved by the genetic algorithms. Then, the reproductive functions that created new individuals would be developed. These functions would use equivalents found in biological reproduction such as crossover and mutation. The simulation, chromosome creator, and reproductive functions would then be combined to form the genetic algorithm. After determining how to modify the chromosomes so that they would be directly used as functional servomotor commands, the evolved pre-programmed sequences could be directly inserted and tested in the physical environment. The simulation parameters such as friction, gravity, and collision forces would have to be carefully calibrated so as to be able to resemble the physical world as closely as possible. Upon completion of physical calibration, the genetic algorithm would be ready to deliver usable pre-programmed inputs.

The secondary goal of this project was to develop a flexible simulation that could be used to work towards a genetic algorithm capable of producing optimal pre-programmed inputs. This

entailed selecting a capable physics simulation package that included features such as the jointed

links that were found on the physical prototype. A robot resembling the physical prototype was

built within the simulation, along with various types of obstacles. Finally, a usable motor input

chromosome and reproductive function was used in concert with the simulation to create a

working genetic algorithm program.

5.2. SIMULATION SELECTION

Open Dynamics Engine (ODE) was selected as the tool for model and simulation. In the

words of the author:

> ODE is a free, industrial quality library for simulating articulated rigid body dynamics -
> for example ground vehicles, legged creatures, and moving objects in VR environments.
> It is fast, flexible, robust and platform independent, with advanced joints, contact with
> friction, and built-in collision detection.[16]

ODE fulfilled the requirements for the simulation tool by providing needed functions such as

rigid body dynamics, frictional approximations, joint structures, and collision detection; the fact

that it was freeware made it all the more appealing. Havok[17] and other commercial simulators

that were investigated for possible use in the project generally ran thousands of dollars to

purchase. The biggest drawback to ODE is that it is not as user-friendly as compared to

consumer physical simulators. First, ODE is still a work in progress. The user manual is far

from finished and modifications to the coded library come out quite frequently. Second, all work

in creating a simulation and model in ODE must be done with in-line programming using C /

C++. Since the benefit of a cost free yet effective physics simulator out-weighed these

negatives, the decision was made to use ODE as the simulator for developing the pre-

programmed controllers.

5.3. SIMULATED ROBOT AND SURROUNDINGS

Having selected the physics simulator, the next step was to use ODE to create an accurate model of the planned physical prototype. This entailed modeling at the simplest level possible and then increasing the complexity of each subsequent design until a fully functional model of the physical prototype was achieved. The final simulated robot built was similar to the actual physical prototype. The parameters of the robot, including number of cars, chassis and wheel size, number of wheels, suspension damping, and maximum torque used by the car links could be easily changed. The wheels were independently powered to give each car differential drive. The two major structural differences were a simplification of the tread system and car links with only one degree of freedom. ODE did not support a tread object that could be used on the robot. Consequently, a system was substituted in its place where each wheel on a side of the robot was powered at the same speed, so as to resemble the tread capabilities as closely as possible. The simulated robot was built so that the number of surfaces on which wheels were mounted could be easily changed. A simple keystroke could select a robot with wheels on the bottom only (figures 16-19), or top and bottom, to give the simulated vehicle a close resemblance to the physical prototype. Whether on top or bottom, each set of wheels on the simulated robot was independently powered as to imitate the abilities of the physical robot. Numerous attempts to get a fully functional powered link with two degrees of freedom failed, because either the links would not work, or the simulation would become unstable. As such, the car links were made so they could only adjust the relative position of the cars in one plane of motion. An interface was designed so that the robot could accept user control of all its joints and wheels. In order to create a realistic test environment, a ramp, a ladder, stairs, and irregular obstacles were designed (see

figures 16-19). The size, position, and location of the obstacles were formatted so as to be easily
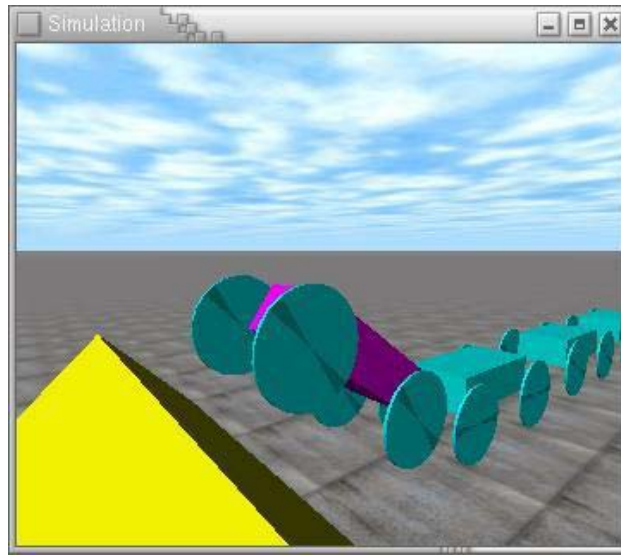
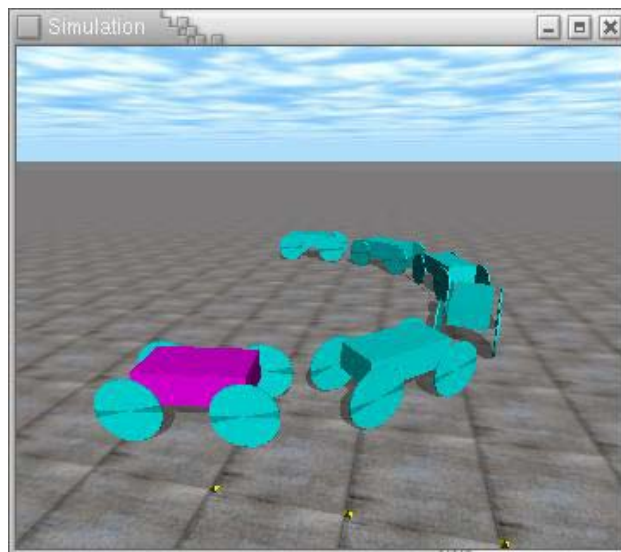modified.
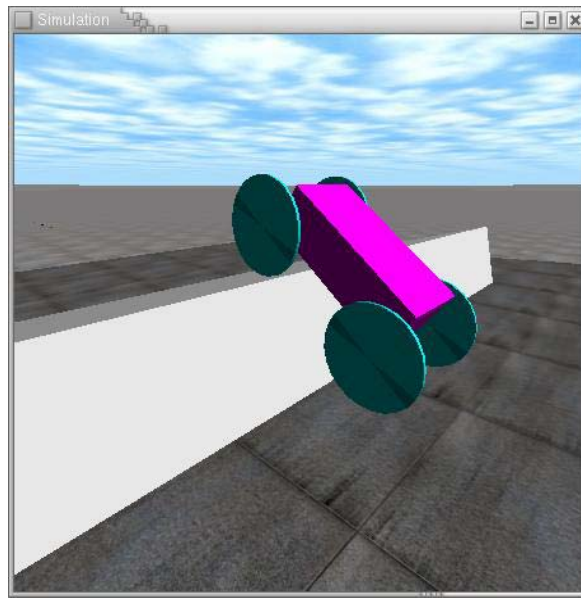
Figure 16: Single Car on Ramp
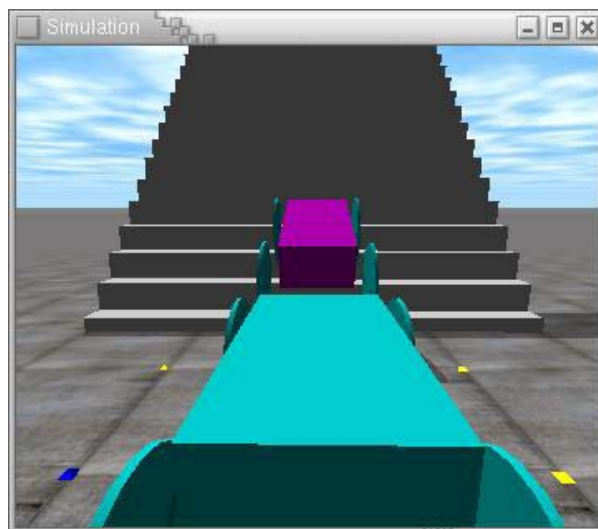
Figure 17: Linked cars in front of stairs

Figure 18: Front car elevated by joint motors (not pictured) to overcome obstacle


Figure 19: Snake robot using powered car links (not pictured) to form shapes

5.4. CHROMOSOME CREATION

Having created a working robot within the simulation, the next step was to create a

format for motor commands that could be used in a genetic algorithm. Since hundreds of these

input matrices (the sequences of motor commands) could be created at a time, it was important to

minimize their memory utilization. The number of motors being used and the amount of time

determined the dimensions of the input matrix. There were two choices for the format of the

actual values (genes) of the chromosome (figure 20). Absolute values were the states at which

the motor was desired. For example, an absolute value input of 45 was a command to elevate

link number one to a position 45 degrees above horizontal. The second choice was to use deltas,

or changes from the joint's previous position. A delta command involved increasing the speed of

a wheel by five units from its previous velocity. The first benefit to using deltas was that they

were a much smaller size. The entire input matrix had values of 1, -1, or 0. Calculating the

position of a joint or the velocity of a wheel at each time step involved multiplying the input

matrix value by a user-defined global variable, and then changing the velocity or position by the

resultant value. In contrast, absolute values would be floating point numbers much larger in size.

The second benefit of the deltas was their ability to be interchanged within the input matrix.

Genetic algorithm reproductive features, such as mutation and crossover, would be far more

effective when randomly swapping increments rather than absolute numbers. For these reasons,

deltas were used as the gene values in the chromosome.
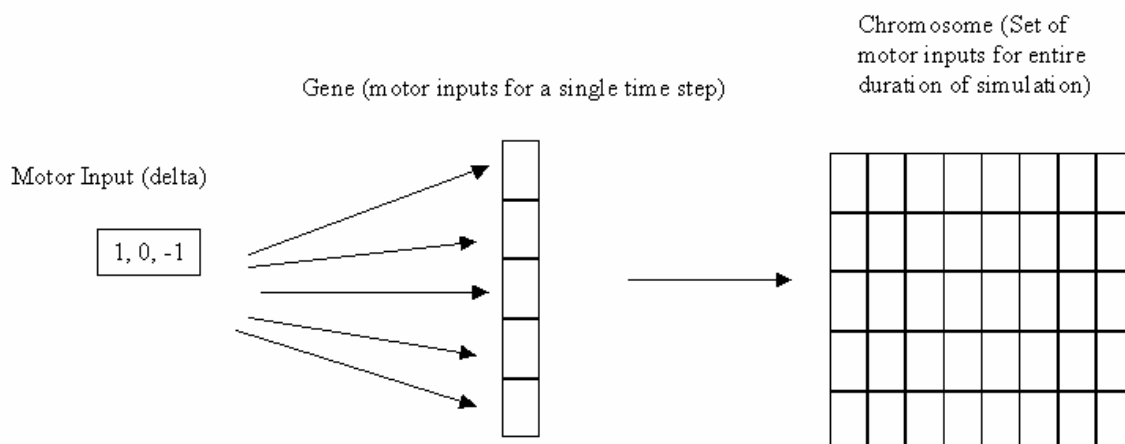
Figure 20: Diagram of chromosome format

5.5. PRELIMINARY GENETIC ALGORITHM

As a culmination of the programming work, a genetic algorithm was built using the simulation and chromosome format. The reproductive function used a tournament-based selection as described by Nolfi.[18] All uses of random numbers in the algorithm involved the selection of a member from a uniformly distributed set of numbers. After the individuals of a generation had all been through the simulation and assigned a fitness, two individuals at a time were randomly picked. If a random number was less than a pre-determined factor, than the individual with the higher fitness was selected for reproduction. If the random number was higher than the pre-determined factor, the other individual was used. One point crossover was the process used to create the new individual. Two individuals from the old generation picked by the tournament selection (the parents), each contributed a piece of their chromosome. The point at which the origin of the new individual's genes changed from one parent to the other was determined by a randomly determined factor. A form of mutation was also introduced to the reproductive function. This consisted of random change to the new chromosomes while they were being created.

The completion of the reproductive functions was the last step to completing a preliminary genetic algorithm capable of evolving motor inputs. In the preliminary testing, the simulated robot was a six-car snake robot, with the cars connected by links that controlled the heading angle of each car. The algorithm attempted to evolve car link motor inputs that would result in forward movement. Consequently, the fitness function used for this problem was the distance moved by the center of mass of the robot in the direction assigned as positive forward movement. A few preliminary tests of the routine indicated the algorithm's ability to improve the average fitness of each generation after each cycle.

6. FUTURE WORK - PROGRAMMING

For the genetic algorithm to produce optimal controllers, two more steps would have to be accomplished. The simulation would need to be fully correlated with the physical environment. Factors such as friction coefficients, density, size, and gravitational force would all have to be determined so that the simulation could match the real world as closely as possible.

The genetic algorithm would then be capable of providing the solution for difficult kinematics and structural problems. Both behavior and structure could be evolved, areas in which Karl Sims has used genetic algorithms successfully in the past. [19],[20] A potential physical problem would involve a robot of six linked cars. Each car link could either control pitch or heading, but not both. Genetic algorithms would be used to determine the optimal joint structure for mobility. Genetic algorithms could also be used to find unusual methods of movement such as pivoting the robot around its center of mass. Eventually, the two efforts of this project could be combined and the genetic algorithms used to develop pre-programmed motions that would enhance the abilities of the physical prototype.

7. CONCLUSION

This project successfully created a working prototype employing structural features designed to enhance the robot's movement capabilities. The spring controlled car link was a unique feature that had never been applied to existing urban search and rescue robot designs. Its ability to control the relative angles between the cars, while at the same time allowing them a degree of compliance, was critical in creating a design that had a significant degree of flexibility. The use of treads on top and bottom of each car was a design feature successfully employed for the first time by the pyramid-exploring robot. The success of the pyramid-exploring robot

validated the concept's ability to improve forward motion capabilities in confined spaces. The user control module demonstrated the ability to create a semi-autonomous system for the robot, in which pre-programmed inputs reduced the complexity of operator control. The combination of unique structural features and simplified control interface resulted in an easy to control robot with a versatile means of mobility.

Unfortunately, the adequate testing needed to validate the effectiveness of this design in an urban search and rescue environment was not performed. Significant delays in hardware acquisition meant that far less time was available for the physical construction efforts. By testing the robot against the three pre-selected terrain types, the robot's ability to move in an urban search and rescue environment could be fully analyzed. Once the robot's capabilities in an urban search and rescue environment were explored, features such as environmental sensors and a long distance tether could then be added to make the robot fully ready to search for survivors in a collapsed building.

The programming efforts provided the first step to a fully functional genetic algorithm. The ODE physics library was successfully used to build a simulated robot and environment that was similar to its physical counterparts. The creation of a similar robot and practical obstacles resulted in a simulation package that had a wide range of uses. This simulation could be used in future work to evolve sequences of motor commands to achieve effective overall movement for segmented robots.

This project created a lightweight, easy to control robot with unique design features such as the spring-servomotor link, and treads on top and bottom of each car. These features gave the robot a large degree of structural flexibility and versatile mobility characteristics. The programming efforts were the first step in using evolutionary programming to develop the

control for such a complicated vehicle. The efforts in building a physical prototype and creating

simulation tools were a major step towards the development of a highly capable urban search and

rescue robot

REFERENCES

[1] Choset, Howard, "Search and Rescue," http://www.ri.cmu.edu/projects/project_407.html; INTERNET. April 1, 2003.

[2] Murphy, Robin, "Biography," http://www.csee.usf.edu/~murphy/ : INTERNET. April 1, 2003

[3] "Center for Robot-Assisted Search and Rescue," http://www.csee.usf.edu/robotics/crasar/ : INTERNET. April 14,2003.

[4] *Packbot: Man Portable Ground Tracked Vehicle.* http://www.irobot.com/rd/p08_PackBot.asp: INTERNET. April 14, 2003

[5] Choset, http://www.ri.cmu.edu/projects/project_407.html

[6] Bogatchev, et al. *Wheel Propulsive Devices for Mobile Robots: Design and Characteristics*. Russian Mobile Vehicle Engineering Institute.

[7] Saito, Masashi, Masakazu Fukaya, and Tetsuya Iwasaki. *Serpentine Locomotion with Robotic Snakes.* IEEE Control Systems Magazine, Feb 2002. p.64-81.

[8] Nilsson, Martin. *Snake Robot, Free Climbing.* IEEE Control Systems Magazine. Vol. 18, No 1. February 1998. p. 21-26.

[9] "Robot Indiana Jones to Penetrate Pyramid," 9-14-02 CNN article. http://www.cnn.com/2002/TECH/science/09/14/pyramids/ : INTERNET. April 14, 2003

[10] Dudek, Jenkin. *Computational Principals of Mobile Robots.* (Cambridge University Press: 2000) p. 19.

[11] Nilsson, Martin. *Snake Robot, Free Climbing.* IEEE Control Systems Magazine. Vol. 18, No 1. February 1998. p. 21-26

[12] Philips, Tony, "Science @ Nasa: Magnetorheological Fluids," http://spaceresearch.nasa.gov/general_info/23aug_MRfluids_lite.html: INTERNET. April 14, 2003.

[13] McComb, Gordon. *The Robot Builder's Bonanza.* (McGraw-Hill Corp: 2001)   p. 308-309.

[14] "Inverse Kinematics," http://freespace.virgin.net/hugo.elias/models/m_ik.htm: INTERNET. April 14, 2003.

[15] Nolfi, Stefano, and Dario Florean. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines.* (MIT Press: 2000) p.22

[16] Smith, Russell. "Open Dynamics Engine," http://www.q12.org/ode/ode.html : INTERNET. April 14, 2003.

[17] "Havok: Real Time Physical Simulation," http://www.havok.com/ : INTERNET. April 14, 2003.

[18] Nolfi, Stefano, and Dario Florean. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines.* (MIT Press: 2000) p.22

[19] Sims, Karl. *Evolving Virtual Creatures.* Computer Graphics. July 1994. pp. 15-22

[20] Sims, Karl. *Evolving 3D Morphology and Behavior by Competition.* Artificial Life IV Proceedings. 1994. pp. 28-39.